Discrete Mathematics 310 (2010) 3303–3309

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

Universal cycles of classes of restricted words

Arielle Leitner [a,1], Anant Godbole [b,*]

[a] Department of Mathematics, University of California, Santa Barbara, CA 93106, United States

[b] Department of Mathematics and Statistics, East Tennessee State University, Box 70663, Johnson City, TN 37614, United States

ARTICLE INFO

Article history:
Received 20 April 2009
Received in revised form 15 July 2010
Accepted 17 July 2010
Available online 25 August 2010

Keywords:
Universal cycle
De Bruijn digraph
Graph connectedness
Equitable word
Ordered Bell number
Password

ABSTRACT

It is well known that Universal cycles (U-cycles) of k-letter words on an n-letter alphabet exist for all k and n. In this paper, we prove that Universal cycles exist for several *restricted* classes of words, including non-bijections, "equitable" words (under suitable restrictions), ranked permutations, and "passwords". In each case, proving the connectedness of the underlying de Bruijn digraph is a non-trivial step.

1. Introduction

The following string has the property that, when wrapped around, it contains all possible words of length three on the binary alphabet {0, 1}: 11100010. We call such combinatorial structures Universal cycles, or U-cycles, since they list all possible "values" of a combinatorial object (in this case binary words of length 3) by the mechanism of a sliding window that wraps around the string. De Bruijn's theorem states that such U-cycles may be constructed no matter what the alphabet size or word length. Chung et al. [2], studied U-cycles for permutations, partitions, and k-sets of an n-set; their work on k-subsets of $[n] := \{1, 2, \ldots, n\}$ was continued by Hurlbert [4], who exhibited the following often quoted U-cycle of 3 subsets of $\{1, 2, \ldots, 8\}$:

1356725 6823472 3578147 8245614 5712361 2467836 7134582 4681258,

where each block is obtained from the previous one by the addition of 5 modulo 8. More recent work is featured in a 2009 issue [9] of *Discrete Mathematics* which contains a selection of papers presented at the Workshop on Generalizations of de Bruijn cycles and Gray Codes, held at the Banff International Research Station, Banff, Canada, December 4–9, 2004.

Jackson [5] studied U-cycles of k-permutations of $[n]$, showing that these exist for $n \geq 3$; $k < n$; the string 123132 provides an example for $n = 3$, $k = 2$. Now k-permutations of $[n]$ may be viewed as *one-to-one* functions from $[k]$ to $[n]$ (or words in which no letter repeats); in the example above, the string "31" can be viewed as the function $f : [2] \to [3]$ for which $f(1) = 3$; $f(2) = 1$. Likewise, the k-letter words on $[n]$ in de Bruijn's theorem are *unrestricted* functions from $[k]$ to $[n]$. Such analogies between words, combinatorial structures, and discrete functions will be drawn and used throughout this paper. We will continue the work begun by Bechel et al. [1], who extended Jackson's work by proving, that for $n \geq 3$,

* Corresponding author. Tel.: +1 423 439 6983; fax: +1 423 439 8361.
E-mail address: godbolea@mail.etsu.edu (A. Godbole).
[1] Tel.: +1 805 893 4079; fax: +1 805 893 2385.

0012-365X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.disc.2010.07.016

U-cycles of *surjections* (onto functions, or words that exhaust the alphabet) from $[k]$ to $[n]$ exist iff $n < k$. For $n = 2$, $k = 3$, an example of a U-cycle of onto functions is given by 112212; for general values of k, n, the length of this U-cycle is $n!S(k, n)$, where $S(k, n)$ are the Stirling numbers of the second kind. Bechel et al. [1] also derived a result on the existence of U-cycles for "1-inequitable" binary functions on $[k]$, (k odd), i.e., binary sequences in which the number of zeros and ones differ by exactly one. The length of such a U-cycle is $2 \cdot \frac{k!}{\lfloor k/2 \rfloor! \lceil k/2 \rceil!}$ and, for $k = 3$, an example is given by 100110.

The following result is basic to the theory of U-cycles; see [10] for a proof.

Theorem 1. *A connected digraph is Eulerian if and only if the in-degree of each vertex is the same as its out-degree.*

Theorem 1 is used to prove the baseline result on U-cycles, namely de Bruijn's theorem:

Theorem 2. *U-cycles of k-letter words on an n-letter alphabet exist for all k, n.*

Proof. We create a digraph, G, with a vertex set that consists of all $k-1$-letter words on the n-letter alphabet. That is, vertices have one less letter than the words which we seek to U-cycle, which will appear as edge labels between vertices as follows: A directed edge is drawn from v_1 to v_2 if the last $k-2$ letters of v_1 are the same as the first $k-2$ letters of v_2, and is labeled with the corresponding concatenated k-letter word. For example, the edge from 11234 to 12344 will be labeled 112344. It is easy to see that the conditions of Theorem 1 are satisfied, and that the Eulerian circuit generates the required U-cycle. □

In this paper, we will use a variation of the above proof in all our results. The key difference is that connectedness of the underlying "de Bruijn digraph" is no longer obvious and will need to be proved.

Jackson's [5] and Bechel et al.'s [1] results are respectively as follows:

Theorem 3. *Let $n \geq 3$. Then a U-cycle of 1–1 functions from $\{1, \ldots, k\} \to \{1, \ldots, n\}$ (and of length $(n)_k = n(n-1)\ldots(n-k+1)$) exists if and only if $n > k$; these are merely permutations of n objects taken k at a time, or, k-letter words on $[n]$ in which no letter repeats.*

Theorem 4. *For $n \geq 3$, a U-cycle of onto functions from $\{1, \ldots, k\} \to \{1, \ldots, n\}$ exists if and only if $k > n$.*

When $k = n \geq 3$, the underlying graph is not connected, and a U-cycle cannot exist. Knuth [7] raised the question of when a U-cycle of one-to-one functions can be explicitly constructed and the first such effort appears to be, for $k = n - 1$, due to Ruskey and Williams [8]. The non-trivial part of the proofs of Theorems 3 and 4 consists of showing the connectedness of the underlying digraph. The same is true for all the results in this paper, where we prove the existence of U-cycles for many more sets of restricted words, such as non-bijections, almost onto words, equitable words, ranked permutations, and "passwords" (we define these later). In each case, we have indicated the easiest proof of connectedness that we could find; the proof of Theorem 8 is the most intricate.

2. U-cycles of non-bijections

Throughout this paper, we will refer to words with entries from an n-letter alphabet $\{1, 2, \ldots, n\}$ as being on $[n]$. A k-letter word on $[n]$, i.e., a function, $f : \{1, \ldots, k\} \to \{1, \ldots, n\}$, is said to be *almost onto* if $|[n] \setminus Range(f)| = 1$. A function $f : [n] \to [n]$ is said to be a non-bijection if $Range(f) \neq [n]$. In any graph, the in-degree, out-degree, and degree of the vertex v are denoted by $i(v)$, $o(v)$, and $\deg(v)$, respectively.

Theorem 5. *A U-cycle of almost onto n-letter words on $[n]$ exists for $n \geq 3$.*

Proof. Since edge labels must correspond to the entities that we wish to form a U-cycle of, namely almost onto words, the vertices of the digraph must be chosen so as to ensure this outcome. On a little reflection, we see that vertices in our digraph will have labels of $M = n-1$-letter words in which at most one letter appears exactly twice and the other letters are distinct. Words with no repeating letters will be called NR words and those with a single repeat will be called OR words. From now on, we will speak of vertices and the words they represent interchangeably. For NR vertices v, $i(v) = o(v) = n-1$, since any letter may be chosen to "complete the edge label" other than the one that does not appear in v; for example with $n = 5$, the vertex 1234 points to 2341, 2342, 2343, and 2344, with the corresponding edge labels being the concatenated almost onto words 12341, 12342, 12343, and 12344. Similarly, the vertices 1123, 2123, 3123 and 4123 each point towards the vertex 1234. Likewise, it is easy to see that for OR vertices with exactly one repeating letter, $i(v) = o(v) = 2$, since any one of the two letters which do not appear in v may be chosen to complete the edge label, which again is an almost onto word. Connectedness is easy to establish for $n = 3, 4$. For $n \geq 5$, we shall exhibit a path from an arbitrary xR vertex to another yR vertex, where x, y may equal N or O.

Case 1. We prove that it is possible to travel from one NR vertex $A = a_1 \ldots a_M$ to another, denoted by $B = b_1 \ldots b_M$. There are two steps in the proof. First we show how we reach a word with the same letters as B, and then exhibit an algorithm by which these letters may be put in the "right order".

Since A and B are both NR, they share all but one letter. Say $a_j \notin B$; $b_i \notin A$. Starting with A, a word with the same letters as B is reached as follows:

$$a_1 \dots a_j \dots a_M \to \cdots \to a_j \dots a_M a_1 \dots a_{j-1} \to a_{j+1} \dots a_M a_1 \dots a_{j-1} a_{j-1} \to a_{j+2} \dots a_M a_1 \dots a_{j-1} a_{j-1} b_i \to$$
$$a_{j+3} \dots a_M a_1 \dots a_{j-1} a_{j-1} b_i a_{j+1} \to \cdots \to a_{j-1} a_{j-1} b_i a_{j+1} \dots a_M a_1 \dots a_{j-3} \to a_{j-1} b_i a_{j+1} \dots a_M a_1 \dots a_{j-2}.$$

Let the word $a_{j-1} b_i a_{j+1} \dots a_M a_1 \dots a_{j-2}$ thus reached be written as $C = c_1 \dots c_M$; B is a permutation of the letters of C. To show that one may travel from C to B it suffices to show that it is possible to go from C to a word created by a single swap of adjacent elements, say $D = c_1 \dots c_{i+1} c_i c_{i+2} \dots c_M$, since any permutation can be attained by a composition of such swaps (this can be accomplished since the underlying Cayley graph is connected. The so-called Johnson–Trotter–Steinhaus order for permutations, see, e.g. [6], gives the stronger condition that the Cayley graph is Hamiltonian). The path from C to D is obtained as follows:

$$c_1 \dots c_i c_{i+1} c_{i+1} \dots c_M \to c_2 \dots c_M c_1 \to \cdots \to c_i c_{i+1} \dots c_M c_1 \dots c_{i-1}$$
$$\to c_{i+1} \dots c_M c_1 \dots c_{i-1} c_{i+1} \to c_{i+2} \cdots c_M c_1 \dots c_{i-1} c_{i+1} c_i \to \cdots \to D.$$

Case 2. To go from an OR word to an NR word, we travel to an intermediate NR word as rapidly as possible and then go from this word to the target NR word as in Case 1.

Case 3. If the goal is to describe a path from A (NR) to B (OR), we identify an NR word C from which B may be reached and then travel from A to C as in Case 1.

Case 4. Traveling between two OR words is done by combining Cases 3 and 4. □

Remark. U-cycles of non-bijections on $[n]$ can be shown to exist using the same argument. We know that for a vertex with no repeats $i(v) = o(v) = n - 1$, since we may travel to or from any of the letters that already appear in the vertex. For a vertex with repeats, $i(v) = o(v) = n$. The corresponding digraph can easily be shown to be connected: a path between two NR words is created as in the proof of Theorem 5, and the other 3 cases are similar; for example when traveling from a word with repeats to an NR word, we first eliminate the repeats from the starting word and then travel to the target NR word.

The results in the rest of the paper are all similar in spirit to Theorem 5. Vertices of the digraph are often of two types, so that we have $i(v) = o(v) = \alpha$ for some vertices and $i(v) = o(v) = \beta$ for other vertices, $\alpha \neq \beta$. Unlike Theorem 2, in which vertices were also "words", albeit of smaller length, we see that in our results vertices might represent different kinds of objects than the edge labels we get by concatenation of the vertex labels. Moreover, as pointed out by a referee, all the U-cycles in this paper have the property that if a word is in the U-cycle then so is any permutation of that word. Thus each of the classes of words is defined only in terms of the symbols contained in each word—and not on the relative order of the symbols within each word. The equality of the in- and out-degrees of the vertices follows from this fact. For this reason we may simply specify the *degree* of each vertex.

3. U-cycles of equitable words

A word is said to be equitable if for all letters i, j, $\|i| - |j\| \leq 1$ where $|i|$ represents the number of times the letter i appears in the word. In other words, a word is equitable if the letters of the alphabet are distributed as equally as possible in the word. In this section, we will prove several theorems about U-cycles on equitable vertices. In Bechel et al. [1], only the binary case was considered. Here too, the nomenclature differed; words in which the numbers of ones and zeros differed by one were called 1-*inequitable*. We prefer to use the terminology of graph labeling.

Theorem 6. *A U-cycle of equitable m-letter words on $[n]$ where $m \equiv 0 \pmod{n}$ does not exist unless $m = n = 2$.*

Proof. Vertices will have $M = m - 1$ letters, and it is clear that $i(v) = o(v) = 1 \forall v$; we must always choose the deficient letter to add on, so that the edge label represents an equitable m-letter word. As a result, we will end up cycling back to the starting word in at most m steps. Thus no U-cycle exists unless $m = n = 2$, when the longest length of the cycle equals the size of the vertex set. □

Example. Here we will take the example of 6-letter words on [4]. Say we have the vertex 11223. This must go to 12233, to preserve equitability. Then we must travel to 22331, 23311, 33112, 31122, 11223. We end up where we started.

Theorem 7. *There exists a U-cycle of equitable m-letter words on $[n]$, where $n < m \equiv 1 \pmod{n}$. The length of the U-cycle is then equal to $n \cdot \frac{m!}{\left(\frac{m-1}{n}\right)!^{n-1}\left(\frac{m-1}{n}+1\right)!}$.*

Proof. Vertices will have $M = m - 1$ letters, where $M \equiv 0 \pmod{n}$. Let $r = \frac{M}{n}$. Since edge labels must represent equitable M-letter words through concatenation of the vertex labels, we see that there must be two types of vertices. In an "equitable" vertex, there will be r occurrences of each letter. Such vertices have $\deg(v) = 2n$, since we can put in any letter to complete the edge label. "Inequitable" vertices will have r occurrences of each of $n - 2$ letters, with one letter appearing $r - 1$ times and the last letter appearing $r + 1$ times. Inequitable vertices necessarily have $i(v) = o(v) = 1$, or $\deg(v) = 2$, since we must travel to the word that makes up for the "deficient" letter.

We will show connectedness by first exhibiting a path from an arbitrary equitable word $A = a_1a_2 \dots a_M$ to another equitable word $B = b_1b_2 \dots b_M$. We add letters of the word B until we are legally able to, i.e., until we reach the inequitable vertex $a_s \dots a_M b_1 \dots b_{s-1}$. Note that at least one of the $r+1$ occurrences of the "overrepresented" letter must be among the a's, since B is equitable and thus has exactly r letters of each kind; consequently $b_1 \dots b_{s-1}$ contains at most r occurrences of the overrepresented letter. Let the first occurrence of the overrepresented letter among the a's occur at a_ℓ. We now add mandatory letters to the inequitable vertex $a_s \dots a_M b_1 \cdots b_{s-1}$ until we reach the equitable word $a_{\ell+1} \dots a_M b_1 \dots b_{s-1} c_1 \cdots c_{\ell-s+1}$. This may now be cycled around to get $c_1 \dots c_{\ell-s+1} a_{\ell+1} \dots a_M b_1 \dots b_{s-1}$, and, finally, we add b_s to get $c_2 \dots c_{\ell-s+1} a_{\ell+1} \dots a_M b_1 \dots b_s$. We repeat the above process until B is reached.

If we wish to exhibit a path from an inequitable A to an equitable B, we first go from A to an equitable C and then from C to B as in the previous paragraph. The last two cases are handled similarly. □

The next result generalizes Theorem 7; notice, however, that the proof is substantially different. Together, Theorems 7 and 8 extend Theorem 3 (which addresses the case when $m < n$).

Theorem 8. *A U-cycle of equitable m-letter words on $[n]$ $(m > n)$ exists whenever $m \equiv k \pmod n$, and $k \neq 0$.*

Proof. Assume that $k \geq 2$, since the $k = 1$ case has already been treated in the previous theorem. Vertices will have $M = m-1$ letters, where $M \equiv k-1 \pmod n$. Let $r = \frac{m-k}{n}$. "Equitable vertices" will have $n-k+1$ letters repeated r times, and $k-1$ letters appearing $r+1$ times. Such vertices will have $i(v) = o(v) = n-k+1$, since we may "add" any letter of which there are r to get an edge label that represents an m-letter equitable word. Vertices which are "inequitable" will have r occurrences of $n-k-1$ letters, $r+1$ occurrences of k letters, and $r-1$ occurrences of a single letter. These vertices will have $i(v) = o(v) = 1$, since we must go to the letter that appears $r-1$ times in order to ensure that the concatenated edge label represents an equitable word.

We will show connectedness by producing a path from an arbitrary equitable word $A = a_1a_2 \dots a_M$ to another equitable word $B = b_1b_2 \dots b_M$; the other three cases will then follow easily. Our strategy will be (i) to first exhibit the fact that we can travel from A to a word B' with the same letter frequencies as B, and then (ii) to show that we can rearrange the letters of B' as needed; the latter will be done through a series of swaps of adjacent elements.

Letters that appear $r-1$, r, and $r+1$ times will be called *deficient*, *normal*, and *super*, respectively. An equitable word has no deficient letters.

(i) Assume that A has super letters labeled $a_1, \dots, a_{k-1}$ and normal letters $a_k, \dots, a_n$. Similarly let B have super letters $b_1, \dots, b_{k-1}$ and normal letters $b_k, \dots, b_n$. The word frequencies of B are clearly obtained by changing some of the super letters in A to normal letters; each such change forces a normal letter to become a super letter. A sequence of such "swaps" permits us to reach the letter frequencies of the target word B. For example, if $A = 112233444555666$ and $B = 111223334445566$, we need to make '6' normal while converting '1' to super status, and make '5' normal while making '3' super at the same time. Suppose we wish to "swap the status" of letters a_i; $1 \leq i \leq k-1$ and a_j; $k \leq j \leq n$ in this fashion. Suppose furthermore that there are $s \geq 1$ super letters that occur before the first occurrence of super letter a_i, and we label these from left to right as $c_1, \dots, c_s$. Finally, suppose that there are $t-s$ normal letters, labeled $d_1, \dots, d_{t-s}$ from left to right, before the first occurrence of a_i. The way in which the c's and d's are intertwined is irrelevant. We replace c_1 by a_j so as to make a_j a super letter; replace each d_i by itself; and restore the super status of $c_1, \dots, c_s$ by replacing c_i by c_{i-1}; $2 \leq i \leq s$ and a_i by c_s. For example we swap the status of '4' and '1' in the word 512625454331466 by proceeding as follows:

$$512625454331466 \to 126254543314661 \to 262545433146615 \to 625454331466152 \to 254543314661521$$
$$\to 545433146615212 \to 454331466152126 \to 543314661521265.$$

If there are no super letters preceding the first a_i, i.e. if $s = 0$, we simply replace each normal letter by itself and the first a_i by a_j. A sequence of such swaps allows us to arrive at B'.

(ii) We now need to be able to reorder the letters of B' in the order desired, i.e., so as to get B. This too will be achieved through a sequence of swaps of adjacent elements. To begin with, however, we show that any equitable word may be "lag cycled" around if we first add a "placeholder", which is any normal letter, to the word. Lag cycling is defined to be a process in which one letter is always missing from the cyclic version of the word, and in which there is an extra letter (the placeholder) that is eliminated at the last step. This initial step makes the cycling legal at all stages, even though an inequitable word may be reached in an intermediate step. More specifically, if we start with an equitable word $A = a_1 \dots a_M$ containing a normal letter x, then the sequence of steps

$$a_1 \dots a_M \to a_2 \dots a_M x \to a_3 \dots a_M x a_1 \to \cdots \to x a_1 \dots a_{M-1} \to A$$

is always legal; for example, if $A = 1122333$, the above sequence might be

$$1122333 \to 1223332 \to 2233321 \to 2333211 \to 3332112 \to 3321122 \to 3211223 \to 2112233 \to 1122333.$$

The reason that the above process works is evident in hindsight: The placeholder creates a lag between the deletion of a letter and its reintroduction. So, e.g., the step $C := a_2 \dots a_M x \to a_3 \dots a_M x a_1 =: D$ is always permissible since by design a_1 is a normal or deficient letter in the word C.

To be able to swap letters, we are going to need *two* placeholders. First note that since there are $k-1$ super letters and $k \leq n-1$, each equitable word *must* have at least two normal letters, denoted by ♡ and ♠, to be used as placeholders.

Assume that we need to swap adjacent letters a_i and a_{i+1}, i.e., go from $a_1 \dots a_i a_{i+1} \dots a_M$ to $a_1 \dots a_{i+1} a_i \dots a_M$. We start by introducing our first placeholder $\heartsuit$ right away, choosing $\heartsuit = a_i$ if a_i is normal, and $\heartsuit \neq a_{i+1}$ if a_i is super:

$$a_1 \dots a_M \to a_2 \dots a_M \heartsuit,$$

and continue lag cycling as in the previous paragraph until we reach

$$a_{i+1} \dots a_M \heartsuit a_1 \dots a_{i-1}.$$

Noting that the above word is equitable, we then introduce the second placeholder $\spadesuit$, *and which should be chosen to be* a_{i+1} *if at all possible.* [Note that a_i cannot be deficient at this stage, so we can choose $\spadesuit \neq a_i$. If a_{i+1} is a normal letter in the word A, we first introduce the other normal letter $\heartsuit$, waiting until this stage to introduce $\spadesuit = a_{i+1}$. Of course if a_{i+1} is a super letter then we cannot have $\spadesuit = a_{i+1}$ at this stage.] We thus get

$$A^* = a_{i+2} \dots a_M \heartsuit a_1 \dots a_{i-1} \spadesuit,$$

and then, in a critical step, transition to

$$a_{i+3} \dots a_M \heartsuit a_1 \dots a_{i-1} \spadesuit a_i.$$

Let us check that this last step is valid. Two things would prevent it from being so. First, a_i could be super, or, second, a_{i+1} could be deficient in the word A^*. Consider the first possibility. Since $\heartsuit \neq \spadesuit$, if neither $\heartsuit$ nor $\spadesuit$ equals a_i, then a_i is either normal or deficient in A^*. If $\spadesuit \neq \heartsuit = a_i$, then a_i must have been normal to begin with and still is. Finally, the possibility $\heartsuit \neq \spadesuit = a_i$ has been ruled out. Can a_{i+1} be deficient in A^*? A review of the possibilities shows that this too is impossible. We thus reintroduce a_i as above into the equitable word A^*, and lag cycle to

$$a_M \heartsuit a_1 \dots a_{i-1} \spadesuit a_i a_{i+2} \dots a_{M-2}.$$

From here, in two steps, and after dropping the first placeholder, we reach

$$A^{**} = a_1 \dots a_{i-1} \spadesuit a_i a_{i+2} \dots a_{M-1} a_M.$$

If $a_{i+1} = \spadesuit$, we are done; the required swap has been achieved. If $a_{i+1} \neq \spadesuit$, a_{i+1} is a normal letter in the equitable word A^{**} and we add another available placeholder $\clubsuit$ to get

$$a_2 \dots a_{i-1} \spadesuit a_i a_{i+2} \dots a_{M-1} a_M \clubsuit,$$

and lag cycle until we reach

$$A^{***} = a_i a_{i+2} \dots a_M \clubsuit a_1 \dots a_{i-1}.$$

Next, we reintroduce the normal letter a_{i+1} into the equitable word A^{***} to yield

$$a_{i+2} \dots a_M \clubsuit a_1 \dots a_{i-1} a_{i+1},$$

go onto

$$a_{i+3} \dots a_M \clubsuit a_1 \dots a_{i-1} a_{i+1} a_i,$$

and lag cycle until the target word

$$a_1 \dots a_{i-1} a_{i+1} a_i \dots a_{M-1} a_M$$

is reached. □

Remark. Suppose we define an *s-inequitable* word as one in which letter frequencies differ by at most s, i.e., $||i| - |j|| \leq s$ for all i, j; $s \geq 1$. (Using this nomenclature, equitable words could be termed 1-*inequitable*, as was done in Bechel et al. [1].) It is then easy to use Theorem 8 to show that a U-cycle of s inequitable m-letter words exists as long as m is not a multiple of n. Here is a sketch of the proof: Assume for simplicity that $s = 2$. Vertices are words of length $M = m - 1$ and may be of three kinds—they may be 1-inequitable (or, equitable), 2-inequitable, or 3-inequitable. For example, if $n = 5$ and $m = 19$, then vertices may have letter frequencies (4, 4, 4, 3, 3), or (4, 4, 4, 4, 2), or (5, 4, 4, 3, 2). In general there are as many cases as there are partitions of the integer $m - 1$ into n parts with the difference between the maximum part size and the minimum part size being at most $s + 1$, and with the minimum size part size having multiplicity one when the above difference equals $s + 1$. In our case $s + 1 = 3$. For these three types of vertices, $i(v) = o(v) = 5, 1, 1$ respectively. To establish connectedness, we first travel from the "start" word to a 1-inequitable one, note than we can backtrack from the target word to another 1-inequitable word, and, finally, go from the first 1-inequitable word thus created to the second as in the proof of Theorem 8.

4. U-cycles of ranked permutations and passwords

Chung et al. [2] suggest investigating U-cycles on tied permutations as a future direction of research. Here we offer one way of defining these, motivated by rankings, seedings, etc. in sports and other events.

Definition. We say that a word satisfies a ranking if the word contains a 1, and if there exists $r \geq 1$ of some letter a, the next letter must be $a + r$.

More informally, these words must follow ordinary rankings in a tournament. For example, the ranking 113 is allowed, but not 112, since the second place is already taken in the tie for the first.

Example. Here are all the rankings allowed on [4]:

111, 122, 221, 212, 113, 311, 131, 123, 312, 231, 321, 132, 213.

We see that a U-cycle of these words exists as follows: 1113212213123. Of the associated vertices, 22, 23, and 32 have $i(v) = o(v) = 1$ and 12, 21, 31, 13, 11 have $\deg(v) = 4$. An enumeration of rankings on $[n]$ for $1 \leq n \leq 18$ (the so-called ordered Bell numbers) may be found as the sequence A000670 in Neil Sloane's website of integer sequences www.research.att.com/~njas/sequences/.

Theorem 9. *A U-cycle of ranked permutations on m-letter words exists for each m.*

Proof. Vertices in the graph consist of $m - 1$-letter words that are consistent with a ranking, i.e., those that can be extended to a ranking. For example, the word 1124 is not consistent with a ranking, but 1135 is. Vertices will either contain a 1 or not. Since all rankings must contain a 1, vertices which do not contain a 1 will have $i(v) = o(v) = 1$, e.g. a vertex such as 2345 will point towards 3451 and the induced edge label will be the ranked permutation 23451. Consider vertices which do contain a 1. Now each vertex will be missing one letter from some ranking. We now ask: how many rankings is such a vertex consistent with? We claim that the answer is 2. Writing a vertex with its letters in non-decreasing order, we see that there is a single "gap" in the ranking, and that gap may always be filled in two ways—by the symbol to the left of the gap or by the position of the gap. Thus $i(v) = o(v) = 2$. For example, say we had 5-letter words, and we had the vertex 2125. This is really the ranking 122x5 where the x may be filled by another 2 (the symbol to the left of the gap) or by a 4 (the position of the gap). This vertex thus points to vertices 1252 and 1254 (the edge labelings are 21252 and 21254) and is pointed at by vertices 2212 and 4212. The vertex 1114557899 has a gap at the end which may be filled with a 9 or an 11.

Connectedness: From any vertex we can travel in the direction of less repeats, until we get to a vertex without any ties. Likewise, we can backtrack from any vertex to one without repeats. It remains to be shown that we can travel from any one vertex A without repeats to another, labeled B, that also has no repeats. Our strategy, distinct from that adopted in previous proofs, will be to show that we may legally travel from A to $O = 111\ldots111$ and then from O to B. Traveling from A to O: Assume, without loss of generality, that the word A has a one. Since A has no repeats, we first replace the '2', it exists, by a '1', transitioning in the process to a word A_1 with two 1s. This is done as follows: First, we add letters to A until the 2 is eliminated. We may now add a 1, and may need to add another 1 in order for the word to have two 1s, as desired. As an illustration, we implement the replacement of 2 by 1 as follows:

$$A \coloneqq 532147 \to 321476 \to 214765 \to 147653 \to 476531 \to 765311 \coloneqq A_1.$$

We next add letters to A_1 until the '3', if it exists, is eliminated, and then add an additional 1 in at most three steps, thus getting a word A_2. Continuing in this fashion, so that when the number j is eliminated, the word is modified to include j ones, we reach O as desired. One strategy we could use for the example started above is the following:

$$A_1 = 765311 \to 653114 \to 531147 \to 311476 \to 114765 \to 147651 \to 476511 \to 765111 =: A_2,$$

and be completed thus:

$$765111 \to 651111 \to 511117 \to 111176 \to \cdots \to 761111 \to 611111 \to$$
$$111117 \to \cdots \to 711111 \to 111111 = O.$$

Traveling from O to B is essentially a reversal of the above process. Given a word B, define B^* to be the word B read backwards. Let P^* be a path from B^* to O created as in the previous paragraph. Then the path P defined to be P^* read backwards, and with entries read backwards too, legally takes us from O to B. For example, to go from O to 741235 (the starting word in the previous paragraph read backwards), we use the steps

$$O \to 111117 \to \cdots \to 111567 \to \cdots \to 113567 \to \cdots \to 567412 \to 674123 \to 741235. \quad \square$$

Definition. We say that an m-letter word on $[n]$ is a *password* if there are $q < n$ distinct classes of symbols in n, and each word must contain at least one element of each class. (Note that the classes need not form a partition of $[n]$.)

More informally, we can think of this as being a security-conscious Internet password that must contain one lower case letter, one number, one symbol, and so on. This general definition of passwords includes more familiar elementary textbook objects such as words with at least one vowel, etc. Note moreover that when $m > n = q$, passwords are onto functions; and when $q = 1$ we get the set up of de Bruijn's theorem. Finally we observe that U-cycles of passwords cannot possibly exist for all values of the parameters—e.g., if $n = m = q$, when passwords are n-permutations of $[n]$. Our next result can almost certainly be improved in several directions, and we invite the reader to do so. For example, Ohad Feldheim, a doctoral student at Tel Aviv University, has shown [3] that the result below is true for $m \geq q + 1$.

Theorem 10. *A U-cycle exists for all m-letter passwords on* $[n]$*, with q distinct classes of symbols, provided that* $m \geq 2q$.

Proof. Vertices will have $M = m - 1$ letters. There are two types of vertices. If a vertex is missing exactly one of the types of symbols, then $i(v) = o(v) = |l_v|$, where $|l_v|$ is the number of the type of missing symbol, l_v, that the vertex is missing. If a vertex has all the different types of symbols, then clearly $i(v) = o(v) = n$.

We will show connectedness by going from an arbitrary word $A = a_1a_2 \ldots a_M$, to a word $B = b_1b_2 \ldots b_M$. As with previous proofs, the main case (among four altogether) is the one where both A and B have representatives from all classes of symbols. We first go from A to A', where A' has one of each type of special symbol as its last q letters. Next, we travel from A' to B', where B' has as its last q letters one of each of the special classes of symbols, but in the same order as they appear in the word B. The word B may now be built without hindrance, one letter at a time. □

Acknowledgements

The research of both authors was supported by NSF REU Grant 0552730, and was conducted at East Tennessee State University in the Summer of 2008, when Leitner was a student at California State University, Chico. The paper has benefited significantly from the comments of two referees, causing improvements in form as well as content.

References

[1] A. Bechel, B. LaBounty-Lay, A. Godbole, Universal cycles of discrete functions, Congr. Numer. 189 (2008) 121–128.
[2] F. Chung, P. Diaconis, R. Graham, Universal cycles for combinatorial structures, Discrete Math. 110 (1992) 43–59.
[3] Ohad Feldheim, Personal communication.
[4] G. Hurlbert, On universal cycles for k-subsets of an n-element set, SIAM J. Discrete Math. 7 (1994) 598–604.
[5] B. Jackson, Universal cycles of k-subsets and k-permutations, Discrete Math. 117 (1993) 114–150.
[6] S.M. Johnson, Generation of permutations by adjacent transposition, Math. Comput. 17 (1963) 282–285.
[7] D. Knuth, The Art of Computer Programming, Volume 4, Fascicle 2, Pearson, NJ, 2005.
[8] F. Ruskey, A. Williams, An explicit universal cycle for the $n - 1$-permutations of an n-set, ACM Trans. Algorithms, 6 (3) (2010) (in press). Article 45.
[9] B. Stevens, G. Hurlbert, B Jackson (Eds.), Discrete Mathematics, vol. 309, 2009 (special issue).
[10] D. West, Introduction to Graph Theory, Prentice Hall, New Jersey, 1996.